

Linzer biol. Beitr.	43/1	97-114	25.07.2011
---------------------	------	--------	------------

## **New Tools for Phylogenetic reconstruction using character state trees**

A. TIEFENBRUNNER, M. TIEFENBRUNNER & W. TIEFENBRUNNER

**Abstract:** From the very beginning of algorithmic supported phylogenetic reconstruction, character state trees were seen as an important tool. Later on this changed due to interest shift to molecular biological data and – we believe – because no simple representation of character state trees within the taxon/character – matrices that are fundamental for any algorithmic phylogenetic reconstruction, were developed.

Here we present a new algorithm for Camin – Sokal parsimony and some new tools that ease computing and simplify the representation of states within the taxon/character – matrix even for very complex character state trees. This method can not only be used for morphological data, but is an aid for the combination of cladograms, developed using e.g. molecular biological data, too. A software that uses these tools is available as freeware from the correspondence author.

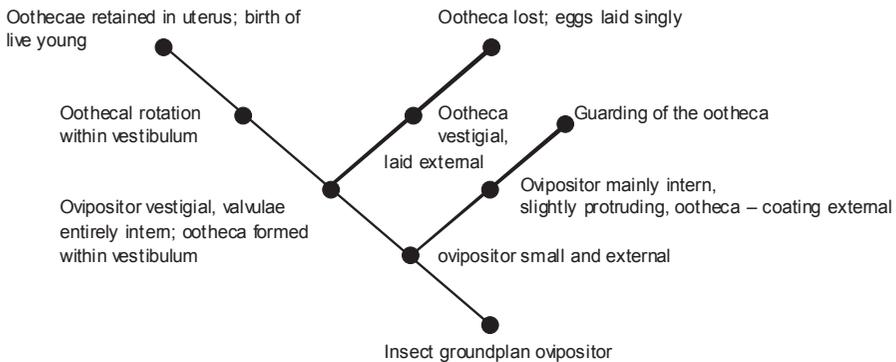
**Key words:** Phylogenetic reconstruction, character state trees, Camin – Sokal parsimony.

### **Introduction**

Only shared derived character states should be used to proof close relationship of taxons (HENNIG 1966). This is the most fundamental principle of "post darwinian", phylogenetic, systematics. It is well known that not to distinguish between primitive (plesiomorph) and derived (apomorph) states, not taking into account the stepwise evolution of them, not to deal with transition series or – if the evolution of states occurred with ramifications – with character state trees (CSTs), will lead to "pre darwinian" systematics that rests upon general similarity and not on phylogenetics. This of course remains true, whether or not the reconstruction is done by computer algorithms.

Taking this into account, it is surprising that some of the most popular methods of "phylogenetic" reconstruction, e. g. "maximum likelihood", "Wagner parsimony" or bayesian approaches, are not able to use existing information concerning the historical order of character states and thus create similarity trees instead of phylogenetic ones. The usage of such methods was criticised repeatedly. Exemplary we cite BERGSTRÖM & XIANGUANG 1998: "Characters used without an understanding of their historical order will most probably distort any cladogram. This is why computer programs for cladograms are very dangerous tools for those who think they have found a shortcut to map phylogeny".

The algorithm supported reconstruction of phylogeny started in accordance with the fundamental principle of phylogenetic systematics, even before Hennig formulated it. The first algorithm for the reconstruction of phylogenetic relationship (a clustering procedure later on named tree popping) was developed by Konrad Lorenz in 1941. He applied it to a species/characters – matrix with 48 characters (mainly behavioural ones) and 20 anatic species. Of course it was not an algorithm for a computer program (which may be the reason why his contribution fell into oblivion) because fast, generally available computers did not exist at this time. Lorenz instead used a wire model to test his algorithm. Most characters were binary, with known historical sequence, one state being primitive relative to the other. This is the simplest possible CST. A state of a character can be primitive relative to another one and at the same time derived to a third. This leads to CSTs that may be very complex, as shown in Fig. 1.



**Fig. 1:** Evolution of the ovipositor and the ootheca of Dictyoptera (Insecta) as an example of a relatively complex character state tree (based on data of GRIMALDI & ENGEL 2005 and EHRMANN 2002). Each node of the tree corresponds to an observed character state.

When CAMIN & SOKAL (1965) started a remarkable experiment to recognize how talented taxonomists successfully reconstruct phylogenetic relationships, they not only invented the "maximum parsimony" method of phylogenetic reconstruction but were also aware of the significance of CSTs. They practically used a simplified version. By the usage of a method developed by SOKAL & SNEATH 1963 for dividing complex CSTs into several binary factors – simple "characters" with only two states where one is more derived than the other – KLUGE & FARRIS 1969 were able to further develop the idea of Camin and Sokal. Nevertheless the coding of the states remained uninformative concerning the question of the relative position of the states in the original CST. No algorithm for the automatic division of a CST into binary factors was presented, so that the creation of complex CSTs was not supported.

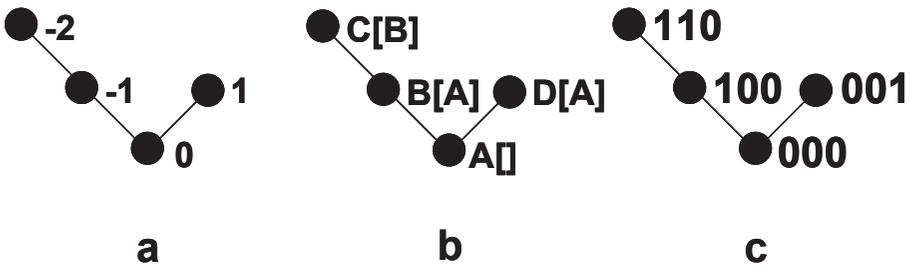
About this time (1968) a coding system that gives information about the relative position of the states in a CST was published by Estabrook but was "only" used for progress in theory, not for practical approaches (e.g. usage in a species/characters – matrix).

Later on the Sankoff algorithm was developed (SANKOFF & CEDERGRÉN 1983) that allowed the usage of CSTs too. Unfortunately using this algorithm, the computational effort increases with the square of the number of states. Therefore only simple CSTs can be utilized.

As a consequence, although phylogenetic reconstruction software exists that deals with linear transition series, to our knowledge currently computer programs that utilize CSTs of any complexity are not available. In this article we present a new algorithm that allows the usage of CSTs of any complexity for the creation of cladograms without necessity to divide them into binary factors and with a computational effort that increases only linear with the number of states. A gratis software that uses this algorithm is available.

### Camin – Sokal parsimony and Character State Trees

CAMIN & SOKAL 1965 used a simple notation to symbolize the structure of a CST and to compute the evolutionary distances between states. Unfortunately it has the disadvantage that only a single bifurcation at the root is possible. The root is labeled with 0, to the left hand the nodes get stepwise decremented by one, to the right hand they get incremented by one, for instance -2, -1, 0, 1 (fig. 2a). This system makes the computation of evolutionary distances between two states very easy. If more than one bifurcation of a CST is desired, this advantage gets lost and additional symbols are necessary, e.g.  $I'$  and  $I''$ . KLUGE & FARRIS 1969 solved this problem by splitting a character into the necessary amount of binary "subcharacters", called "factors" (fig. 2c). Of course, the necessity to split does not enhance the creation of complex CSTs.



**Fig. 2a-c:** The same character state tree with Camin – Sokal (a) and Matrioshka (b) coding, and (c) as a combination of three subcharacters (factors), each one with two states.

Another way to deal with CSTs of any complexity is to use the Sankoff algorithm (SANKOFF & CEDERGREN 1983) for non-uniformly weighted states (FELSENSTEIN 2004). The cost for a substitution from a primitive to a more derived state depends on the distance between the two states on the CST, the cost for a substitution in the opposite direction is infinite. For the example used in fig. 2, using the notation of 2a, the cost matrix would be:

**Tab. 1:** Cost matrix for Camin-Sokal parsimony using the Sankoff algorithm and the CST of fig. 2a.

		to descendant			
		1	0	-1	-2
from ancestor	1	0	$\infty$	$\infty$	$\infty$
	0	1	0	1	2
	-1	$\infty$	$\infty$	0	1
	-2	$\infty$	$\infty$	$\infty$	0

Using the Sankoff algorithm the computational effort increases with the square of the

number of character states of a CST (WILLIAMS & FITCH 1990). Once again this does not inspire the creation of complex CSTs. What we need is a simple algorithm with the features that the computational effort increases arithmetically with the number of character states. To reach this aim we further require

- a basic notation to describe a CST, and
- elementary tools for calculation.

### • Matrioshka – operator

To symbolize the structure of a CST, we use an operator that connects two states. It is a pointer that points from one state at its immediate ancestral neighbour. We call it "matrioshka – operator".

For example, in fig. 2b *C* points at *B*:  $C[B]$ . The immediate primitive state is written in brackets, which of course is an arbitrary convention. The name of the operator refers to the famous Russian doll that contains a doll that contains a doll ... and so on. The reason becomes obvious if we present *C* in this way:  $C[B[A[[]]]]$ . This is called the **path** from *C* to the root. To get knowledge of the whole structure of the tree for each state we only need to know its immediate more primitive neighbour. Thus this operator is appropriate to be used within the taxon / character – matrix (Tab.2).

**Tab. 2:** Example of a species / characters – matrix with matrioshka – coding. In character 4 there are more states in the character state tree than species in the matrix. Because the description of the character state tree within the matrix must be complete, here it is necessary to connect more than two states with the Matrioshka operator. Redundancy within the matrix is allowed (in character 3 the connection "*state3[state2]*" appears two times, although the second time "*state3*" would give enough information), but not necessary (characters 2 and 4). The root of the CST points at nowhere, the brackets remain empty.

	Char 1	Char 2	Char 3	Char 4
Species 1	A[]	0[]	State1[]	D[B[A[[]]]]
Species 2	B[A]	0	State2[State1]	E[B]
Species 3	C[B]	0	State3[State2]	F[C[A]]
Species 4	D[A]	1[0]	State3[State2]	G[C]

For the creation of a CST there are some rules that must be valid for Matrioshka – states:

1. Uniqueness of the root. There is exactly one state that points at nowhere, the root of the CST. If the most primitive state is for instance A, we write A[].
2. Prohibition of self – reference. No state points at itself: A[A] is forbidden.
3. Prohibition of cyclic reference. A[A] is a special case of a loop connection. Cyclic references are generally forbidden, e.g. B[A] and A[B].
4. Uniqueness of the reference. A state cannot point at more than one other state: C[A, B] is forbidden.
5. Contrary to rule 4 it is of course allowed that any number of states point at one state, as long as rules 1 to 4 are not violated. Within a CST from one node any number of branches can sprout (see also ESTABROOK 1968).

- **Matrioshka - set**

Our primary tool for calculations is the matrioshka – set. The elements it includes are all the states we visit, if we, starting from a state, go to the root (all states that belong to the path from a state to the root). As an example let us take  $C$  from fig. 2b. The matrioshka – set  $\mathbf{m}_C$  associated to  $C$  is:  $\mathbf{m}_C = \{A, B, C\}$ , for  $D$ ,  $\mathbf{m}_D = \{A, D\}$ . For  $A$ ,  $\mathbf{m}_A = \{A\}$ . There is no empty matrioshka – set.

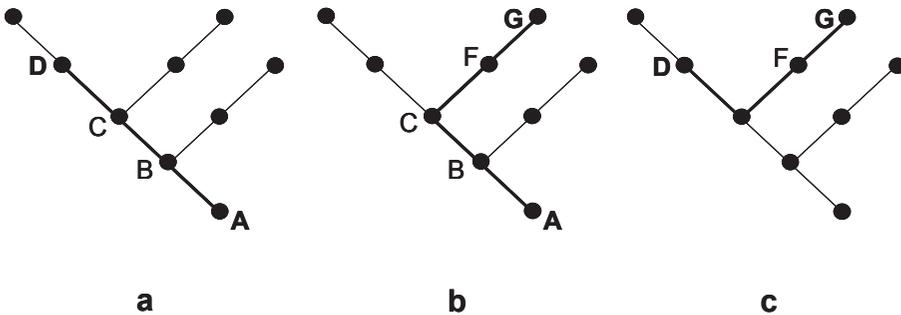
The intersection of two Matrioshka – sets has all characteristics of a matrioshka – set, too. This means that there exists a state that is an element of this set which has all the other elements of the set as ancestral states. Furthermore all its ancestral states – without any exception – are elements of the set. The state that has this characteristics is the element with the highest matrioshka – value.

- **Matrioshka – value**

The matrioshka – value of a state is the quantity of elements of its matrioshka – set. The matrioshka – value of  $C$ ,  $|\mathbf{m}_C| = 3$ ; of  $D$ ,  $|\mathbf{m}_D| = 2$ ; of  $A$ ,  $|\mathbf{m}_A| = 1$  (fig. 2b).

### Evolutionary distance of two character states

As an example we want to compute the distance of the character states  $D$  and  $G$  of the CST of fig. 3 (which has the same structure as the one of fig. 1). To do this we need the path from  $D$  to the root  $A$  (fig 3a) and the path from  $G$  to the root (fig. 3b). The distance is the number of states that belong either to the path shown in fig. 3a or to the one shown in 3 b, but not to both. The states that fulfil this criterion can be seen in fig. 3c as dark circles.



**Fig. 3a-c:** A CST with the same structure as the one of fig. 1. The paths to the root of the states  $D$  (3a) and  $G$  (3b) are accentuated, as well as the elements of the set  $S_{DG}$  (fig. 3c, see eq. 1).

Because all the states that belong to the path from a state to the root are elements of the Matrioshka set of this state, too, to compute the evolutionary distance  $d_{AB}$  of any two character states  $A$  and  $B$  we may define a set  $S_{AB}$ :

$$\text{Eq. 1) } S_{AB} = (\mathbf{m}_A \cup \mathbf{m}_B) - (\mathbf{m}_A \cap \mathbf{m}_B)$$

Then  $d_{AB}$  is  $|S_{AB}|$ , the quantity of elements of  $S_{AB}$ .

In our example, fig. 3:  $\mathbf{m}_D = \{A, B, C, D\}$ ;  $\mathbf{m}_G = \{A, B, C, F, G\}$  and  $S_{DG} = \{DFG\}$ . Therefore  $d_{DG} = 3$ .

If we follow the above instructions, we count how many branches at the CST are separating two states. This is our definition of "evolutionary distance".

### Cladogram (Taxon tree)

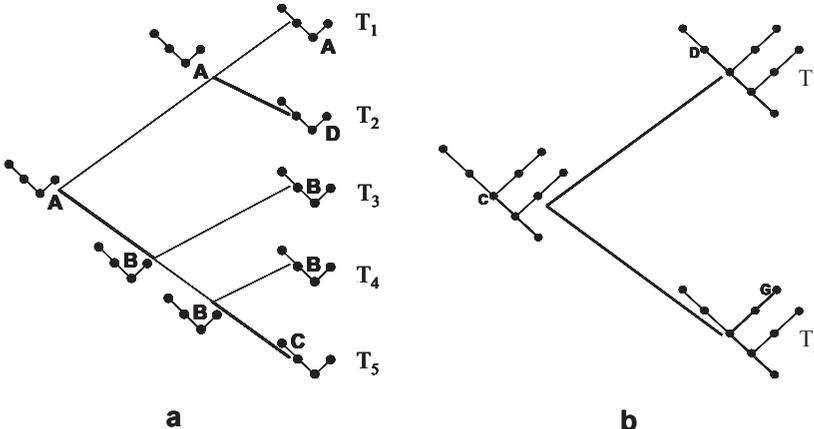
A cladogram is a reconstruction of the evolution from the last common hypothetical ancestor to the taxons, of which we want to know the relationship. So each cladogram is a hypothesis about the relationship of any taxons (e. g. species). To simplify the following explanations, here we discuss the topology of cladograms. Its fundamental structure is that of a binary dendrogram. It consists of:

1. **nodes** ( $n$  **tips**, which represent existing taxons and  $n-1$  inner nodes that symbolize hypothetical taxons)
2.  $2(n-1)$  **branches** that symbolize the path of evolution from an existing or hypothetical taxon and – in the direction of the root – a second one that represents its ancestor (see also 3). The length of a branch is the evolutionary distance of the nodes that are connected by the branch.
3. a **bifurcation** consists of three nodes and two branches. Two of the nodes lie in the direction of the tips (they can be tips). The converging branches connect them with the third one, the origin, that lies in the direction of the root. The origin represents the immediate, common ancestor.
4. The **root** is the hypothetical common ancestor of all nodes of the cladogram.

Orientation: to distinguish cladograms and CSTs easily, we draw cladograms from left (root) to right (tips) and CSTs from bottom (root) to top.

### Quality of the reconstruction of a cladogram

It is necessary to estimate the quality of the reconstruction of the evolutionary course the cladogram represents, so that we can choose a better in favour of a worse one. Our criterion of quality is the global quantity of necessary transitions from one character state to another. The less "evolutionary steps" in the whole cladogram are necessary, the better, which means that we are searching for the most parsimonious hypothesis (principle of economy, CAMIN & SOKAL 1965).



**Fig. 4a-b:** (a) Cladogram for five species (the tips T1 to T5) and the character which is coded in fig. 2b as CST. The inner nodes of the cladogram are hypothetical species. Their character state is reconstructed by the algorithm described in the text. Prominent branches denote an evolutionary step (a change of state). (b) Bifurcation of a cladogram with the character coded in fig. 3.

Fig. 4a gives an example of a cladogram for five species. The character states of the interior nodes are already reconstructed. It is easy to see that for each bifurcation those states are accentuated in the CST of the bifurcation origin (dark circles) that belong to both accentuated paths of the CSTs of the two nodes which lie in the direction of the tips. Thus, if O ist the state of the origin, and A and B are the states of the tipwards nodes, it follows:

$$\text{Eq. 2) } {}^m\text{O} = ({}^m\text{A} \cap {}^m\text{B})$$

This has the consequence that the reconstructed state of the origin is not necessarily identical to the state of one of the tipwards nodes. Take as an example the CST of fig. 3 and the bifurcation of a cladogram (fig. 4b). If the tipwards nodes of any bifurcation of a cladogram have the states D and G, respectively, following eq. 2 we would assign the state C to the origin.

Using matrioshka – sets, the algorithm that allows us to count the number of necessary transitions for any evolutionary course and thus for any cladogram, is very simple to calculate. The procedure needs two steps:

For each character: starting with those bifurcations where the both right nodes are tips, we firstly create the matrioshka – sets of the states that are related to them. Next we calculate the intersection of these matrioshka - sets. The result is related to the basal node, the origin. Next we take the bifurcations where the Matrioshka sets of both right nodes are already known – they are either the calculation result of a right bifurcation or a single tip. In doing so we move leftwards until we reach the bifurcation that has the root as origin.

Using these sets we determine the length of each branch. As already mentioned the length is the evolutionary distance of the nodes that are connected by the branch. We can compute it with the aid of equation 1 using the Matrioshka – sets that were related to the nodes in step one.

We have to do this for all branches of the cladogram and then have to sum up the results. Furthermore we must sum up the results for each character to reach our final aim, the quality criterion of the cladogram.

The computational effort of this algorithm increases linearly with the number of character states.

### **Reconstruction of the course of evolution**

During step 1 we associated a matrioshka set to each node of the cladogram. The element of this set, which has the highest matrioshka – value is the state that must be associated to the node.

## **Zusammenfassung**

Bereits die allerersten Versuche, durch die Verwendung von Algorithmen zu phylogenetischen Rekonstruktionen zu kommen, benützten Merkmalsbäume als wichtiges Hilfsmittel. Mit der Interessenverlagerung hin zu molekularbiologischen Daten veränderte sich das, weil bei diesen die Unterscheidung zwischen "primitiveren" und "abgeleiteteren" Merkmalen meist nicht getroffen werden kann. Außerdem existierte zu diesem Zeitpunkt keine einfache Möglichkeit Merkmalsbäume in Taxon/Merkmal - Matrizen zu repräsentieren.

Hier beschreiben wir einen neuen Algorithmus für "Camin-Sokal-Parsimony" (Prinzip der Sparsamkeit) und einige neue Verfahren, die einerseits der Berechnungsvereinfachung dienen, andererseits die Darstellung von Merkmalsausprägungen innerhalb einer Taxon/Merkmal - Matrix auch für beliebig komplexe Merkmalsbäume ermöglichen. Gegenwärtig ist diese Methode natürlich vor allem für morphologische Merkmale interessant, könnte sich aber auch für Cladogramme als nützlich erweisen, die kombinierte Datensets, z. B. auch molekularbiologische Daten, verwenden. Eine Software, die die neuen Verfahren benützt, ist als Freeware beim Korrespondenzautor erhältlich.

### Literature

- BERGSTRÖM J. & H. XIANGUANG (1998): Chengjiang arthropods and their bearing on early arthropod evolution. — In: EDGECOMBE G.D. (ed.), *Arthropod fossils and phylogeny*, Columbia University Press, 151-184.
- CAMIN J.H. & R.R. SOKAL (1965): A method for deducing branching sequences in phylogeny. — *Evolution* **19**: 311-327.
- EHRMANN R. (2002): *Mantodea: Gottesanbeterinnen der Welt*. — Münster: NTV, 519 Seiten.
- ESTABROOK G.F. (1968): A general solution in partial orders for the Camin-Sokal model in phylogeny. — *Journal of theoretical biology* **21**: 421-438.
- FELSENSTEIN J. (2004): *Inferring phylogenies*, Sinauer Associates, Inc., Publishers Sunderland Massachusetts, 664 pp.
- GRIMALDI D. & M.S. ENGEL (2005): *Evolution of the insects*, Cambridge University Press, 755 pp.
- HENNIG W. (1966): *Phylogenetic Systematics*. — University of Illinois Press, Urbana.
- KLUGE A.G. & J.S. FARRIS (1969): Quantitative phyletics and the evolution of anurans, *Systematic Zoology* **18**: 1-32.
- LORENZ K. (1941): Vergleichende Bewegungsstudien an Anatinen, Beitrag einer Festschrift für Oskar Heinroth zum 70. Geburtstag. — *Journal für Ornithologie* **89**, Ergänzungsband III: 194-293.
- MEACHAM C. (1981): A manual method for character compatibility analysis. — *Taxon*, **30**: 591-600.
- SANKOFF D. & R.J. CEDERGREEN (1983): Simultaneous comparison of three or more sequences related by a tree. — In: SANKOFF D. & J.B. KRUSKAL (ed.), *Time warps, string edits and macromolecules: the theory and praxis of sequence comparison*, Addison-Wesley: 253-263.
- SOKAL R.R. & P.H.A. SNEATH (1963): *Numerical Taxonomy*. — W.H. Freeman, San Francisco.
- WILLIAMS P.L. & W.M. FITCH (1990): Finding the minimal change in a given tree. — In: DRESS A. & A. von HAESLER (ed.), *Trees and hierarchical structures*, Springer Verlag, 137 pp.

Anschriften der Verfasser: Astrid TIEFENBRUNNER  
Martin TIEFENBRUNNER  
Logistic Management Service, Rosenstrasse 7,  
80331 Munich, Germany

Wolfgang TIEFENBRUNNER  
Bundesamt für Weinbau,  
Gölbeszeile 1, 7000 Eisenstadt, Austria  
E-Mail-Korrespondenz: [w.tiefenbrunner@bawb.at](mailto:w.tiefenbrunner@bawb.at)

## Appendix I

As a commemorative publication to the 70<sup>th</sup> birthday of the famous ornithologist Oskar Heinroth, in 1941 Konrad Lorenz, one of the founders of ethology and nobel prize winner of 1973, published a voluminous article to show that behavioural patterns can be genetically fixed and hence can be as well used in systematics, as morphological characters. This proof was immediately recognized as an important one. In the same publication, more or less by the way, Lorenz invented a method how to create a phylogenetic tree by the usage of a species/characters-matrix (48 mainly behavioural characters from twenty Anatid species, fig. 1). This method is now known as tree popping and was reinvented (in a very derived version) 40 years later by Meacham (MEACHAM 1981).

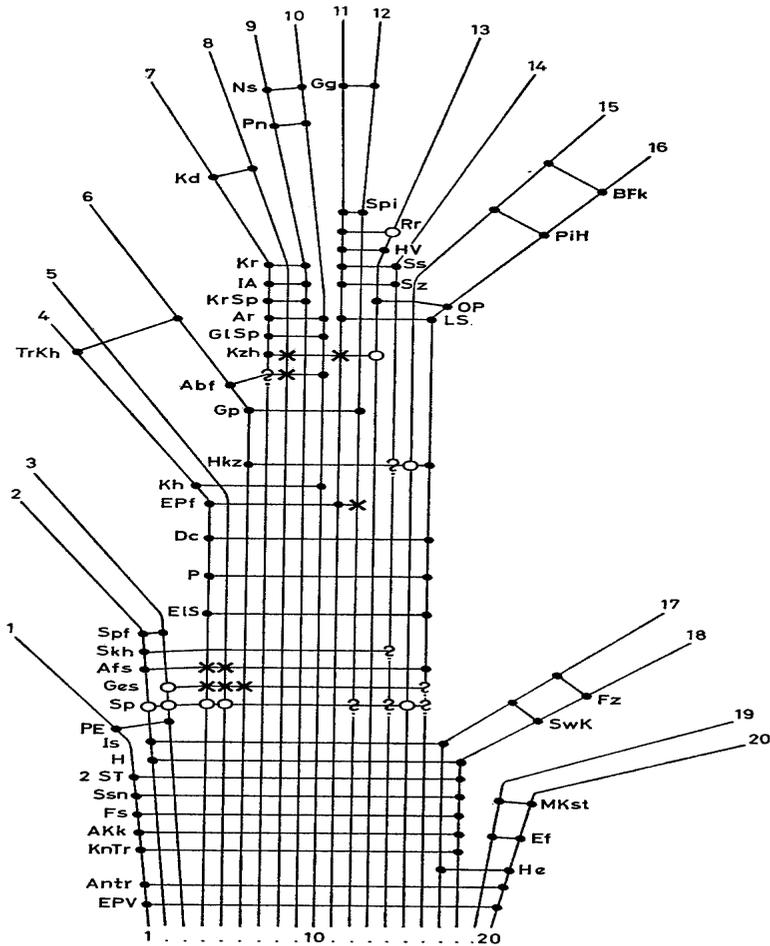


Fig. 1 (from LORENZ 1941): Resulting tree out of a species/characters-matrix of 48 mainly behavioural characters from twenty Anatid species (in fact, this graphic is matrix and tree in one).

In the publication of Lorenz 1941, most of the characters were binary. His algorithm started with a sorting process. Those characters, where most of the species showed the derived state, were used first. The species that showed the derived state were bundled together. Lorenz of course did not use a computer in 1941, instead he used for each species a vertically oriented, thick wire. Those species that shared a derived state were bundled with a horizontally oriented, thin wire that represents a character. Character by character or state by state, respectively, the reconstruction, fig. 1, occurred.

The lines with letters define the characters (for character explanation see LORENZ (1941)), the numbers define the species. Horizontal lines: common, derived character states. As can be seen, only derived states are used for clustering (with two exceptions). Vertical lines are leading to species. Crosses: missing characters; circles: special differentiation of the characters, question mark: lack of knowledge. The species names are from LORENZ (1941) as follows:

1: *Cairina moschata*, 2: *Lampronessa sponza*, 3: *Aix galericulata*, 4: *Mareca sibilatrix*, 5: *Mareca penelope*, 6: *Chaulelasmus strepera*, 7: *Nettion crecca*, 8: *Nettion flavirostre*, 9: *Virago castanea*, 10: *Anas* spp., 11: *Dafila spinicauda*, 12: *Dafila acuta*, 13: *Poecilonetta bahamensis*, 14: *Poecilonetta erythrorhyncha*, 15: *Querquedula querquedula*, 16: *Spatula clypeata*, 17: *Tadorna tadorna*, 18: *Casarca ferruginea*, 19: *Anser* spp., 20: *Branta* spp.

The names of 1, 3, 10, 17, 19 and 20 are valid. The valid names of the others are: 2: *Aix sponza*, 4: *Anas sibilatrix*, 5: *Anas penelope*, 6: *Anas strepera*, 7: *Anas crecca*, 8: *Anas flavirostris*, 9: *Anas castanea*, 11: *Anas georgica spinicauda*, 12: *Anas acuta*, 13: *Anas bahamensis*, *Anas erythrorhyncha*, 15: *Anas querquedula*, 16: *Anas clypeata*, 18: *Tadorna ferruginea*.

Fig. 1 is a tree and a species/characters – matrix in one. Thus we can transform fig. 1 into a species/characters – matrix with matrioshka – coding (tab. 1) and use it to create a Camin – Sokal – parsimony cladogram (fig.2) with the aid of our software (PYRE Classic for PhYlogenetic REconstruction using character state trees).

Because Lorenz gave in his article a very accurate description of the characters, it is possible to connect the simple characters to more complex character state trees. Instead of 48 characters, we get only 18 (tab. 2a and 2b). We can use tab. 2b for Camin – Sokal reconstruction too (fig. 3). Because less characters have to be used, the software arrives at a result earlier than with the data of tab. 1. The result is of course virtually the same (if the data transformation would be perfect, it would be the same. In some cases the text of the article and fig. 1 are contradictory). Furthermore fig. 2 leads more or less to the same result than fig. 1.

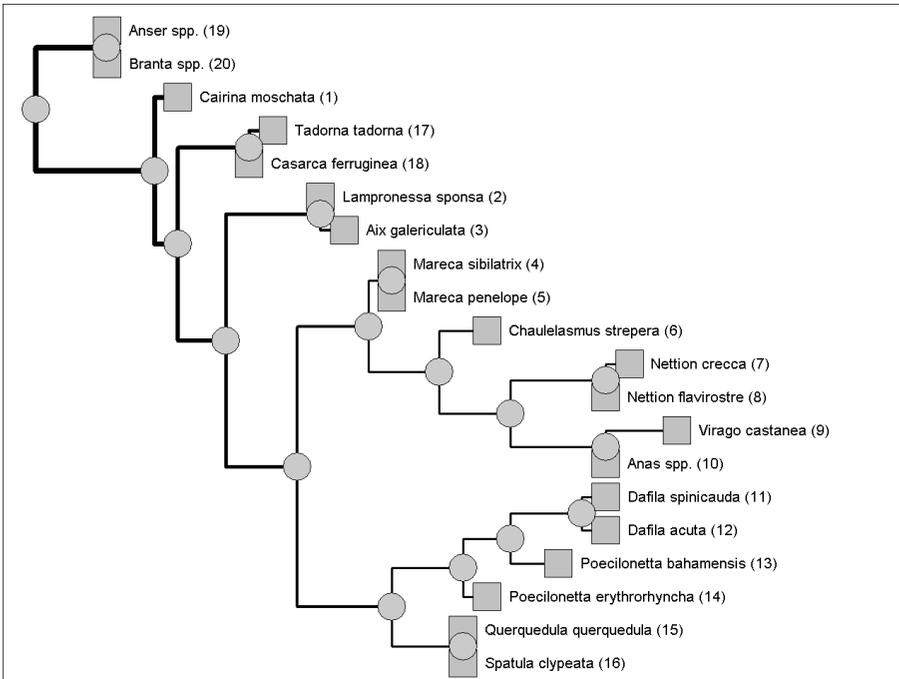


Fig. 2: Result of a Camin-Sokal parsimony phylogenetic reconstruction using the data from tab. 1.

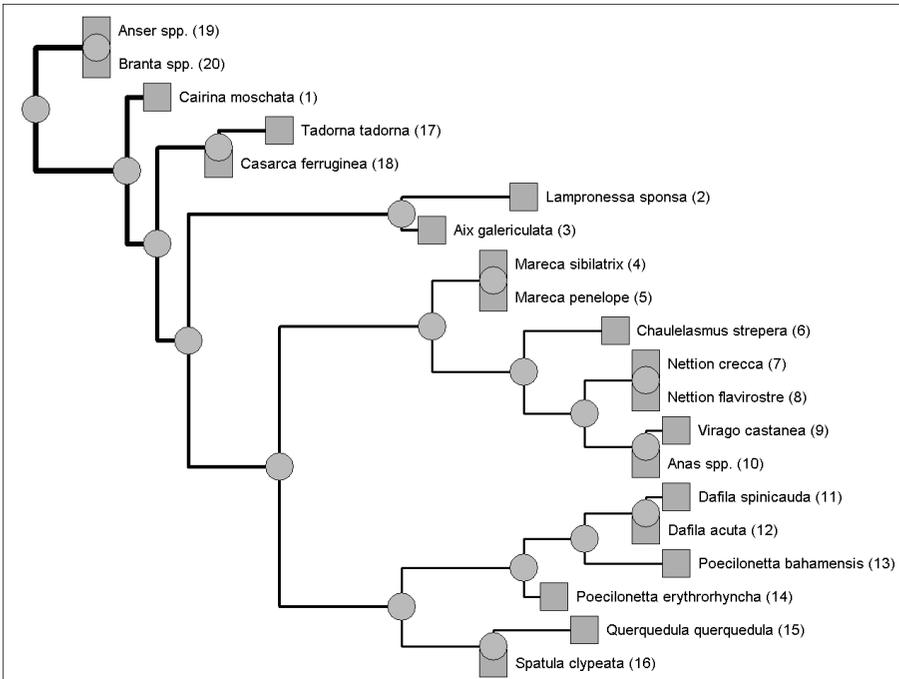


Fig. 3: Result of a Camin-Sokal parsimony phylogenetic reconstruction using the data from tab. 2a.



Tab. 1: Species/characters – matrix out of fig. 1., continuation.

Species / characters	KrSp	Kzh	LS	MKstf	Ns	OP	P	PE	PIH	Pn	Rr	Skh	Sp	Spf	Spi	Ss	Ssn	SwK	Sz	TrKh
<i>Aix galericulata</i> (3)	0	0	0	0	0	0	0	1 0	0	0	0	1 0	1 0	1 0	0	0	1 0	0	0	0
<i>Anas</i> spp. (10)	0	1 0	0	0	1 0	0	0	1 0	0	1 0	0	1 0	1 0	0	0	0	1 0	0	0	0
<i>Anser</i> spp. (19)	0	0	0	1 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<i>Branta</i> spp. (20)	0	0	0	1 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<i>Cairina moschata</i> (1)	0	0	0	0	0	0	0	1 0	0	0	0	0	0	0	0	0	0	0	0	0
<i>Casarca ferruginea</i> (18)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1 0	0	0
<i>Chaulelasmus strepera</i> (6)	0	0	0	0	0	0	1 0	0	0	0	0	1 0	1 0	0	0	0	1 0	0	0	1 0
<i>Dafila acuta</i> (12)	0	1 0	1 0	0	0	0	1 0	0	0	0	1 0	1 0	1 0	0	1 0	1 0	1 0	0	1 0	0
<i>Dafila spinicauda</i> (11)	0	0	1 0	0	0	0	1 0	0	0	0	1 0	1 0	1 0	0	1 0	1 0	1 0	0	1 0	0
<i>Lampronessa sponosa</i> (2)	0	0	0	0	0	0	0	1 0	0	0	0	1 0	1 0	1 0	0	0	1 0	0	0	0
<i>Mareca penelope</i> (5)	0	0	0	0	0	0	1 0	0	0	0	0	1 0	1 0	0	0	0	1 0	0	0	1 0
<i>Mareca sibilatrix</i> (4)	0	0	0	0	0	0	1 0	0	0	0	0	1 0	1 0	0	0	0	1 0	0	0	1 0
<i>Nettion crecca</i> (7)	1 0	1 0	0	0	0	0	1 0	0	0	0	0	1 0	1 0	0	0	0	1 0	0	0	0
<i>Nettion flavirostre</i> (8)	1 0	0	0	0	0	0	1 0	0	0	0	0	1 0	1 0	0	0	0	1 0	0	0	0
<i>Poecilonetta bahamensis</i> (13)	0	1 0	1 0	0	0	1 0	1 0	0	0	0	1 0	1 0	1 0	0	0	1 0	1 0	0	1 0	0
<i>Poecilonetta erythrorhyncha</i> (14)	0	0	1 0	0	0	1 0	1 0	0	0	0	1 0	1 0	1 0	0	0	1 0	1 0	0	1 0	0
<i>Querquedula querquedula</i> (15)	0	0	1 0	0	0	1 0	1 0	0	1 0	0	0	?	1 0	0	0	1 0	1 0	0	1 0	0
<i>Spatula clypeata</i> (16)	0	0	1 0	0	0	1 0	1 0	0	1 0	0	0	0	1 0	0	0	1 0	1 0	0	0	0
<i>Tadorna tadorna</i> (17)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<i>Virago castanea</i> (9)	1 0	1 0	0	0	1 0	0	1 0	0	0	1 0	0	1 0	1 0	0	0	0	1 0	0	0	0

**Tab. 2a** : Species/characters – matrix. The characters of tab. 1 are connected to more complex character state trees.

Species / characters	I	II	III	IV	V	VI	VII	VIII	IX	X	XI	XII	XIII	XIV	XV	XVI	XVII	XVIII
<i>Aix galericulata</i> (3)	A 1	B A	0 1	0 1	A 0 1 1	D	A 0 1	C A	B A	0 1	B A	0 1	A 0 1	A 1	0 1	0 1	0 1	1 0 1
<i>Anas</i> spp. (10)	B A	B A	1 0 1	1 0 1	B 0 1	B	C B 1	A 0 1	C B 1	A 0 1	C A 1	B A 1	B A 1	A 1	0 1	0 1	0 1	1 0 1
<i>Anser</i> spp. (19)	D A	0 1	0 1	0 1	C 0 1	A	0 1	0 1	A 1	0 1	A 1	0 1	0 1	A 1	0 1	0 1	0 1	0 1
<i>Branta</i> spp. (20)	D A	0 1	0 1	0 1	C 0 1	A	0 1	0 1	A 1	0 1	A 1	0 1	0 1	A 1	0 1	0 1	0 1	0 1
<i>Cairina moschata</i> (1)	A 1	0 1	0 1	0 1	A 0 1 1	A 1	0 1	0 1	B A	0 1	A 1	0 1	A 0 1	A 1	0 1	0 1	0 1	1 0 1
<i>Casarca ferruginea</i> (18)	A 1	A 0 1	0 1	0 1	C 0 1	A	0 1	0 1	B A	0 1	A 1	0 1	A 0 1	A 1	1 0 1	0 1	0 1	1 0 1
<i>Chaulelasmus strepera</i> (6)	C B	B A	1 0 1	0 1	B 0 1	C B 1	C B 1	B A 1	C B 1	A 0 1	C A 1	0 1	A 0 1	A 1	0 1	0 1	0 1	1 0 1
<i>Dafila acuta</i> (12)	A 1	C B 1	1 0 1	0 1	B 0 1	A 1	C B 1	A 0 1	F C 1	A 0 1	C A 1	0 1	A 0 1	C B 1	0 1	1 0 1	1 0 1	1 0 1
<i>Dafila spinicauda</i> (11)	A 1	C B 1	1 0 1	0 1	B 0 1	B	C B 1	A 0 1	F C 1	A 0 1	C A 1	0 1	A 0 1	C B 1	0 1	1 0 1	1 0 1	1 0 1
<i>Lamprolaima sponsa</i> (2)	A 1	B A	0 1	0 1	A 0 1 1	D C 1	E B 1	B A 1	A 0 1	A 0 1	B A 1	0 1	A 0 1	A 1	0 1	0 1	0 1	1 0 1
<i>Mareca penelope</i> (5)	C B	B A	1 0 1	0 1	B 0 1	B	B A 1	B A 1	C B 1	0 1	A 1	0 1	A 0 1	A 1	0 1	0 1	0 1	1 0 1
<i>Mareca sibilatrix</i> (4)	C B	B A	1 0 1	0 1	B 0 1	B A 1	B A 1	B A 1	C B 1	0 1	A 1	0 1	A 0 1	A 1	0 1	0 1	0 1	1 0 1
<i>Nettion crecca</i> (7)	B A	B A	1 0 1	0 1	B 0 1	B A 1	C B 1	A 0 1	E D 1	A 0 1	C A 1	A 0 1	B A 1	A 1	0 1	0 1	0 1	1 0 1
<i>Nettion flavirostre</i> (8)	B A	B A	1 0 1	0 1	B 0 1	B	C B 1	A 0 1	E D 1	A 0 1	C A 1	A 0 1	B A 1	A 1	0 1	0 1	0 1	1 0 1
<i>Poecilonetta bahamensis</i> (13)	A 1	D C 1	1 0 1	0 1	B 0 1	B	B A 1	A 0 1	G F 1	A 0 1	C A 1	0 1	A 0 1	B A 1	0 1	1 0 1	1 0 1	1 0 1
<i>Poecilonetta erythrorhyncha</i> (14)	A 1	A 0 1	1 0 1	0 1	B 0 1	A 1	B A 1	A 0 1	G F 1	A 0 1	C A 1	0 1	A 0 1	B A 1	0 1	1 0 1	1 0 1	1 0 1
<i>Querquedula querquedula</i> (15)	A 1	E A 1	1 0 1	0 1	B 0 1	E A 1	B A 1	A 0 1	G F 1	B A 1	C A 1	0 1	A 0 1	A 1	0 1	1 0 1	0 1	1 0 1
<i>Spatula clypeata</i> (16)	A 1	E A 1	0 1	0 1	B 0 1	E	B A 1	0 1	G F 1	A 0 1	C A 1	0 1	A 0 1	A 1	0 1	1 0 1	0 1	1 0 1
<i>Tadorna tadorna</i> (17)	A 1	A 0 1	0 1	0 1	C 0 1	A 1	D A 1	0 1	B A 1	0 1	A 1	0 1	A 0 1	A 1	1 0 1	0 1	0 1	1 0 1
<i>Virago castanea</i> (9)	B A	B A	1 0 1	1 0 1	B 0 1	B	C B 1	A 0 1	D C 1	A 0 1	C A 1	B A 1	B A 1	A 1	0 1	0 1	0 1	1 0 1

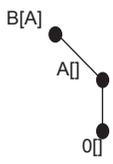
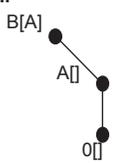
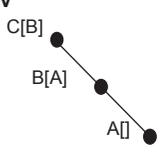
**Tab. 2b:** Description of the new character state trees.

	Species	Characters
<p><b>I</b></p> <pre> graph TD     Root(( )) --- C[B]((C[B]))     Root --- B[A]((B[A]))     Root --- A[ ]((A[ ]))     C[B] --- B[A]     B[A] --- D[A]((D[A]))                     </pre>	<p>A: 1-3, 11-18                      B: 7-10                      C: 4—6                      D: 19,20</p>	<p>2ST                      MkSt                      Kh                      TrKh</p>
<p><b>II</b></p> <pre> graph TD     Root(( )) --- D[C]((D[C]))     Root --- C[B]((C[B]))     Root --- A[0]((A[0]))     C[B] --- B[A]((B[A]))     B[A] --- A[0]     A[0] --- E[A]((E[A]))     A[0] --- 0[ ]((0[ ]))                     </pre>	<p>0: 1,19,20                      A: 14, 17, 18                      B: 2-10                      C: 11, 12                      D: 13                      E: 15, 16</p>	<p>H                      PiH                      Skh                      Hv                      Rr</p>
<p><b>III</b></p> <pre> graph TD     Root(( )) --- A[0]((A[0]))     Root --- 0[ ]((0[ ]))                     </pre>	<p>III 0: 1-3, 17-20                      A: 4-16                      IV 0: 1-8, 11-20                      A: 9, 10</p>	<p>DC                      Ns</p>
<p><b>IV</b></p> <pre> graph TD     Root(( )) --- A[0]((A[0]))     Root --- 0[ ]((0[ ]))                     </pre>		
<p><b>V</b></p> <pre> graph TD     Root(( )) --- A[ ]((A[ ]))     Root --- B[ ]((B[ ]))     Root --- C[ ]((C[ ]))     A[ ] --- P1(( ))     B[ ] --- P1     P1 --- 0[ ]((0[ ]))     C[ ] --- 0[ ]                     </pre>	<p>V: A: 1-3                      B: 4-16                      C: 17-20</p>	<p>HE                      PE                      P</p>
<p><b>VI</b></p> <pre> graph TD     Root(( )) --- D[C]((D[C]))     Root --- C[B]((C[B]))     Root --- A[ ]((A[ ]))     C[B] --- B[A]((B[A]))     B[A] --- A[ ]     A[ ] --- E[A]((E[A]))                     </pre>	<p>A: 1,12, 14, 17-20                      B: 4, 5, 7-11, 13                      C: 6                      D: 2, 3                      E: 15, 16</p>	<p>Antr                      Sp                      Bfk                      Spf</p>

**Tab. 2b:** Continuation

	Species	Characters
<p><b>VII</b></p> <pre> graph TD     O["O[]"] --- A["A[]"]     A --- B["B[A]"]     A --- E["E[A]"]     B --- C["C[B]"]     B --- D["D[B]"]         </pre>	<p>O: 1, 18-20                      A: 3                      B: 4, 5, 13-16                      C: 6-12                      D: 2                      E: 17</p>	<p>IS                      EIS                      Gg</p>
<p><b>VIII</b></p> <pre> graph TD     O["O[]"] --- A["A[]"]     A --- B["B[A]"]     A --- C["C[A]"]         </pre>	<p>O: 1, 16-20                      A: 7-15                      B: 2, 4-6                      C: 3</p>	<p>Ges</p>
<p><b>IX</b></p> <pre> graph TD     A["A[]"] --- B["B[A]"]     B --- C["C[B]"]     B --- A2["A[]"]     C --- D["D[C]"]     C --- E["E[D]"]     D --- F["F[C]"]     D --- G["G[F]"]         </pre>	<p>A: 19, 20                      B: 1-3, 17-18                      C: 4-6, 10                      D: 9                      E: 7,8                      F: 11, 12                      G: 13-16</p>	<p>KnTr                      Epf                      Kr                      Kd                      Gg                      OP</p>
<p><b>X</b></p> <pre> graph TD     O["O[]"] --- A["A[]"]     A --- B["B[A]"]         </pre>	<p>O: 1, 3-5, 17-20                      A: 2, 6-14, 16                      B: 15</p>	<p>Hkz</p>
<p><b>XI</b></p> <pre> graph TD     A["A[]"] --- B["B[A]"]     A --- C["C[A]"]         </pre>	<p>A: 1, 4, 5, 17-20                      B: 2, 3                      C: 6-16</p>	<p>Afs</p>

Tab. 2b: Continuation

	Species	Characters
<b>XII</b> 	0: 1-6, 11-20 A: 7, 8 C: 9,10	Ar Pn
<b>XIII</b> 	0: 19-20 A: 1-6, 11-18 B: 7-10	GlSp
<b>XIV</b> 	A: 1-10, 15-20 B: 13, 14 C: 11, 12	Ss Spi
<b>XV</b> 	0: 1-16, 19,20 1: 17, 18	Fz
<b>XVI</b> 	0: 1-10, 17-20 1: 11-16	LS
<b>XVII</b> 	0: 1-10, 15-20 1: 11-14	Sz
<b>XVIII</b> 	0: 19, 20 1: 1-18	Ssn

# ZOBODAT - [www.zobodat.at](http://www.zobodat.at)

Zoologisch-Botanische Datenbank/Zoological-Botanical Database

Digitale Literatur/Digital Literature

Zeitschrift/Journal: [Linzer biologische Beiträge](#)

Jahr/Year: 2011

Band/Volume: [0043\\_1](#)

Autor(en)/Author(s): Tiefenbrunner Astrid, Tiefenbrunner Martin, Tiefenbrunner Wolfgang

Artikel/Article: [New Tools for Phylogenetic reconstruction using character state trees 97-114](#)